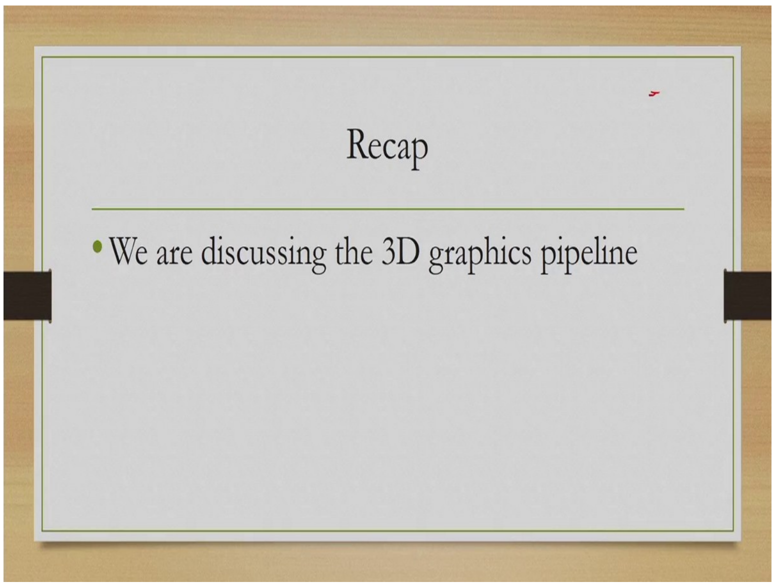
**Computer Graphics**
**Doctor Samit Bhattacharya**
**Department of Computer Science and Engineering**
**Indian Institute of Technology Guwahati**
**Lecture 6**
**Various Boundary Representation Techniques**

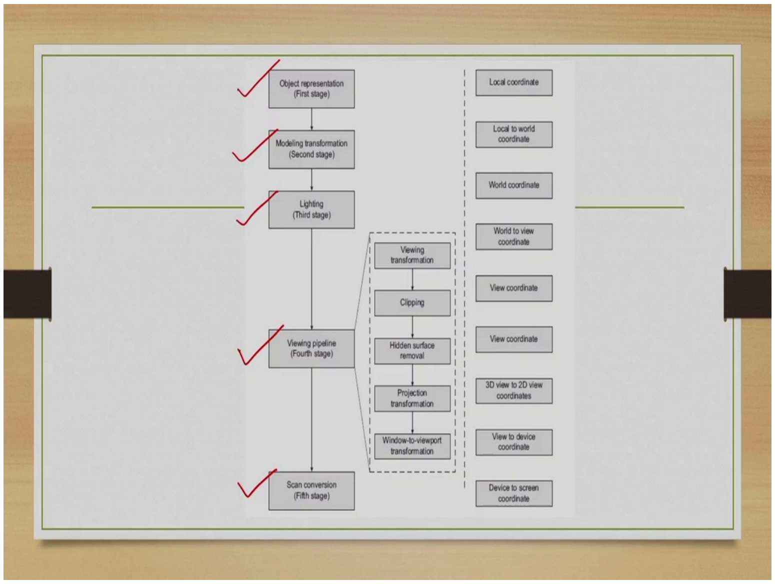
Hello and welcome to lecture number six in the course, computer graphics.

(Refer Slide Time: 0:38)




So, we started our discussion on 3D object representation, which is the first stage of the graphics pipeline.

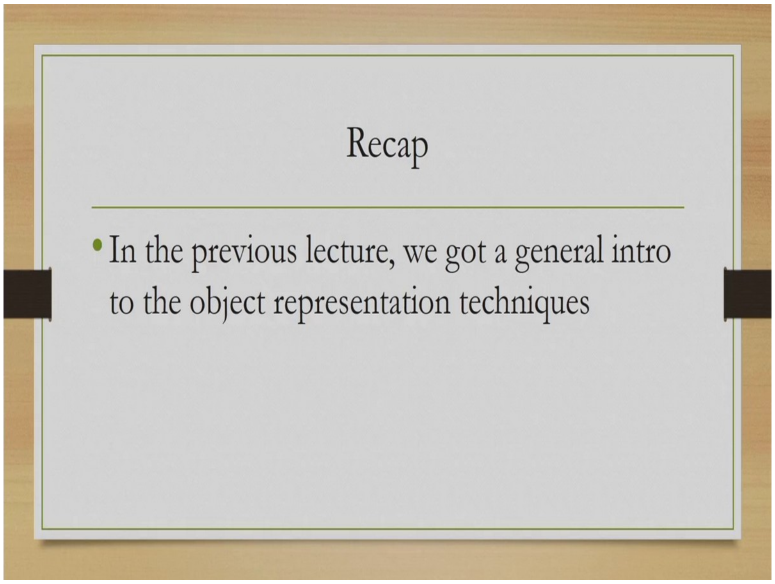
(Refer Slide Time: 0:49)




To recap, let us see the pipeline again. There are 5 broad stages. As being shown on this screen, first stage is object representation, which we are currently discussing, the other stages we will take up in subsequent lectures, namely the modelling transformation, lighting, viewing pipeline and scan conversion.

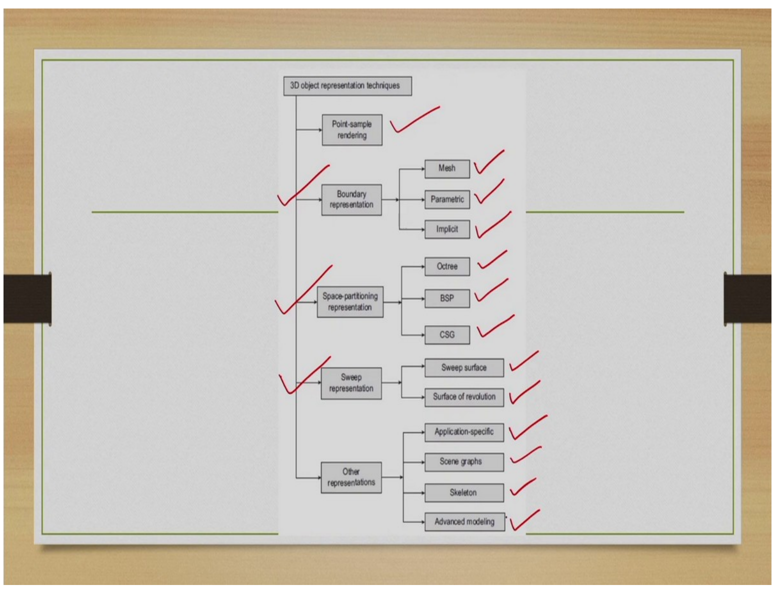
One point I would like to mention here is that although, in this course I will follow the pipeline stages in the way shown here, in practice, it is not necessary to have this exact sequence. Some stages may come after some other stages. For example, lighting may be done after viewing pipeline or in between some of the transformations of viewing pipeline and so on. So, the sequence that I am showing here need not be followed exactly during implementation of a graphics system. This is just for our understanding of the stages involved and the sequence may vary.

(Refer Slide Time: 2:18)




Now, what we have learned in the previous lecture, we got a general introduction to various object presentation techniques.

(Refer Slide Time: 2:27)

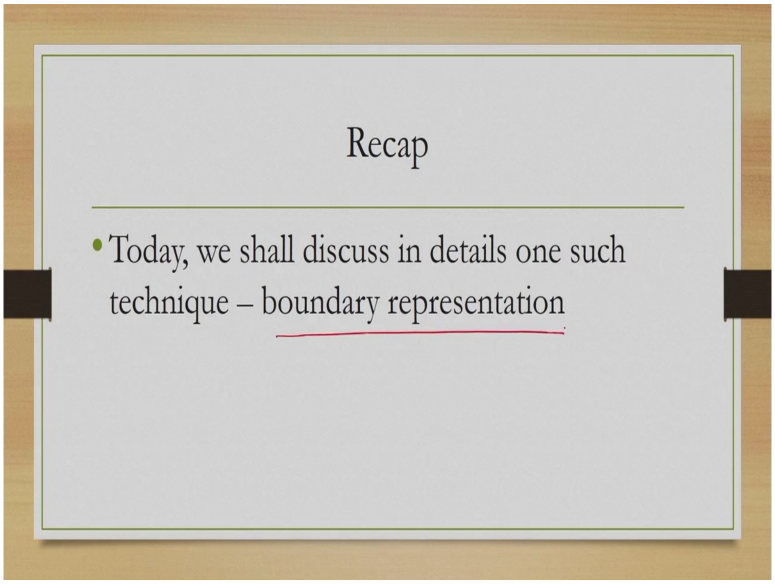



What were those techniques that we discussed? One technique is point sample rendering, then we have boundary representation technique, space partitioning techniques and sweep representation technique. These are the 4 broad categories we mentioned, each of which has subcategories boundary representation, has three subcategories; mesh representation, parametric representation and implicit representation.

Space partitioning has three subcategories; octree representation, BSP representation and CSG representation. BSP stands for binary space partitioning, whereas CSG stands for computational solid geometry. In sweep representation, we have two techniques; sweep surfaces and surface of revolution.

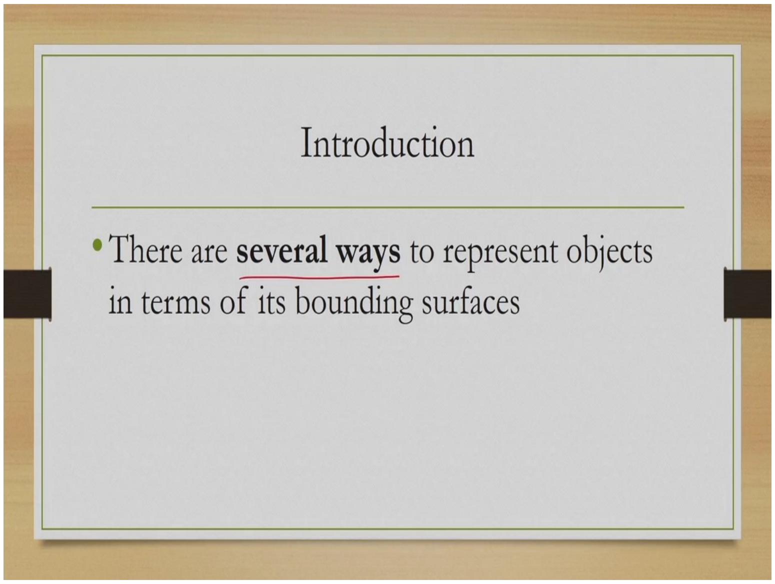
Apart from these 4 broad categories, we have other representations as well. Some are application specific, there are some general advanced representation techniques, namely scene graphs, skeleton models, skeletal models and advanced modelling techniques. Now, in the advanced modelling techniques we have many such techniques, fractal representation, points sample rendering, particle systems and so on.

(Refer Slide Time: 3:56)



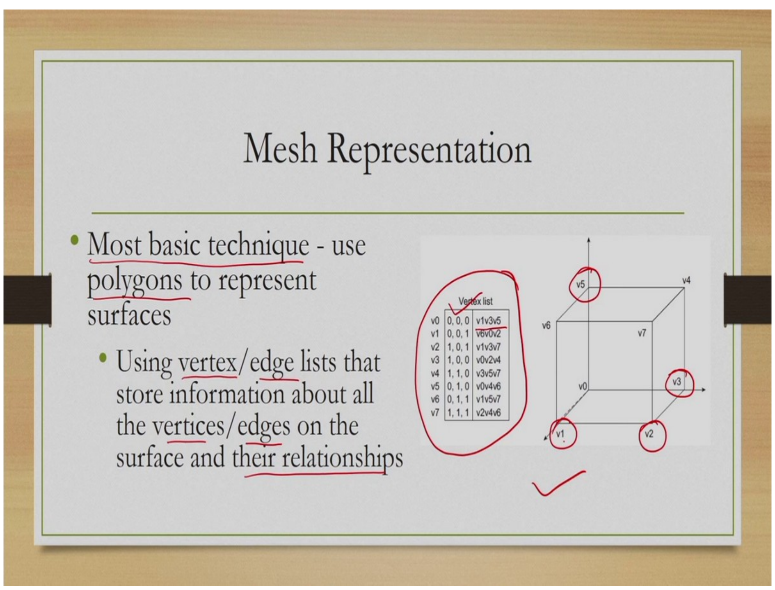

Today, we shall discuss in details one of those techniques, namely boundary representation techniques. We already have seen that in boundary representation techniques we represent an object in terms of its bounding surfaces or the surfaces that constitutes its boundary. Now, those surfaces can be simple polygons or complex steps.

(Refer Slide Time: 4:31)




There are several ways to represent these bounding surfaces. We mentioned three subcategories of representation; mesh representation, implicit representation and parametric forms. So today we will get introductory idea to all these three representation techniques.

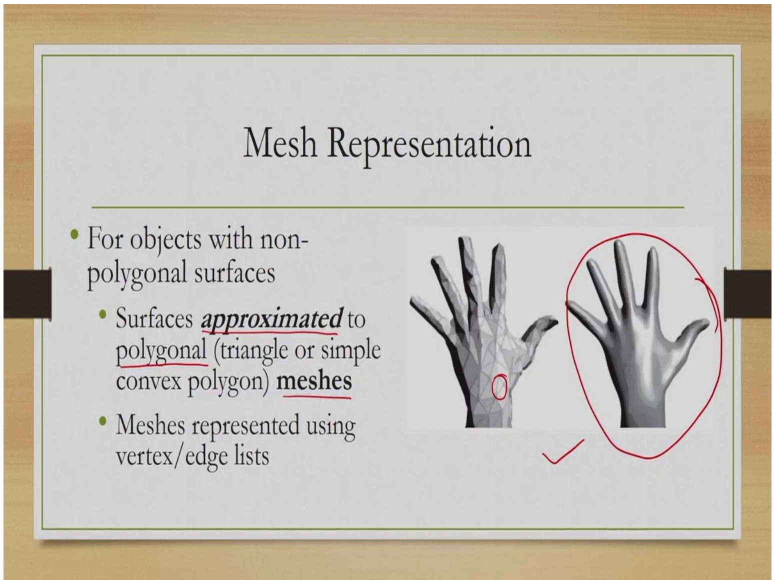
(Refer Slide Time: 5:00)




Let us start with the mesh representation. This is the most basic technique of representing objects in a scene, where we use polygons to represent the surfaces. Now the polygons in terms are represented using vertex or edge lists that store information about all the vertices or edges of the surface and their relationship.

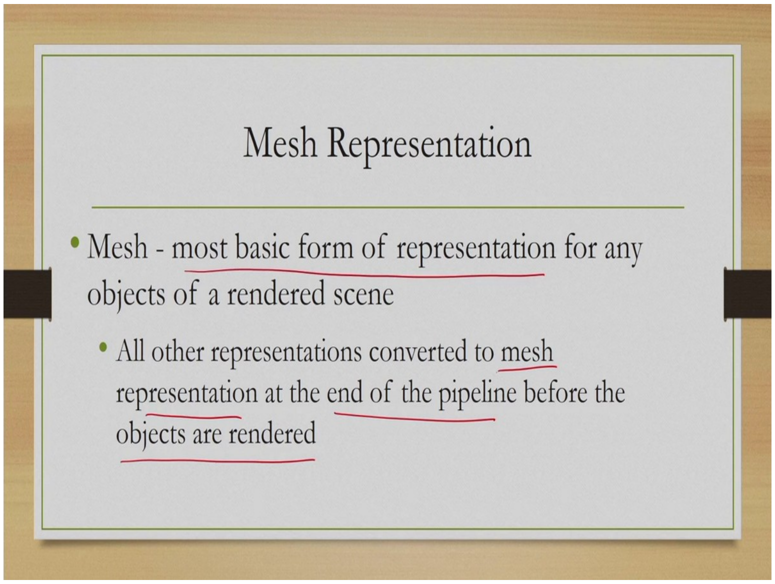
For example, consider the figure here, you are representing a cube in terms of its vertices v1, v2 and so on up to v7, so there are 8 vertices. And this one is the representation where we are storing the vertices with coordinate values and some other values, capturing the relationships. For example, here in this first row, what it tells is that v0 is connected to v1, v3 and v5. Similarly, each vertex stores the other vertices which, it has connection to, this is one representation, there can be other ways to represent it.

(Refer Slide Time: 6:32)



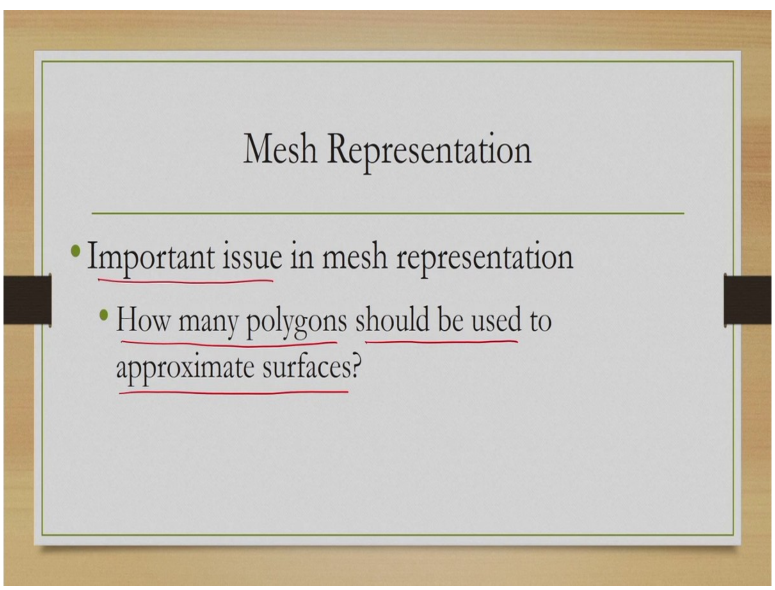

Now, sometimes the surfaces need not be polynomial, but in mesh representation, what we can do is we can approximate anything to polygonal meshes like the figure shown here, here, this hand actually does not contain any polygonal surface. But this hand surface I can approximate with this type of triangular meshes where lots of triangles are used to approximate it. And again, these meshes are represented using vertex and edge lists.

(Refer Slide Time: 7:27)



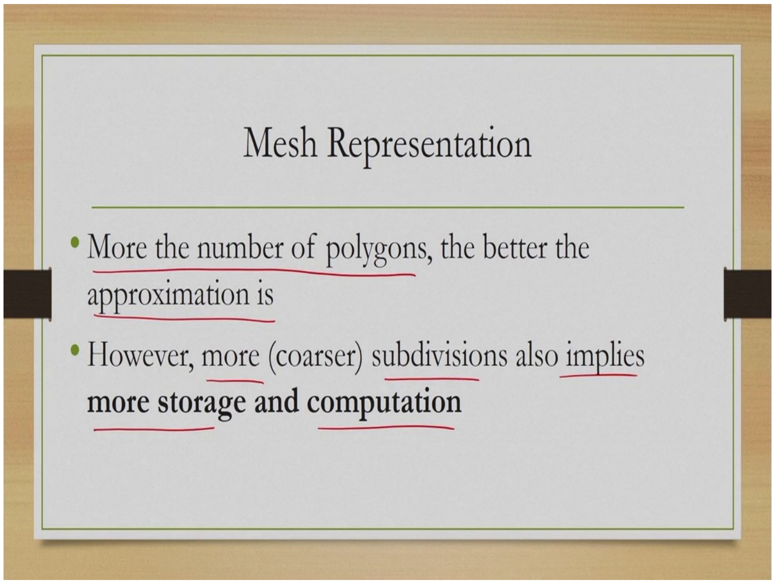

In fact, the mesh representation is most basic form of representation any other representation that we may use will ultimately be converted to mesh representation at the end of the pipeline before the objects are rendered. So, we have to keep this in mind. So, whatever representation we use and we will learn about in subsequent discussions, at the end, everything is converted to a mesh representation.

(Refer Slide Time: 8:02)




Now there is one important issue. That is how many polygons should we use to approximate the surfaces? That is a very fundamental question.

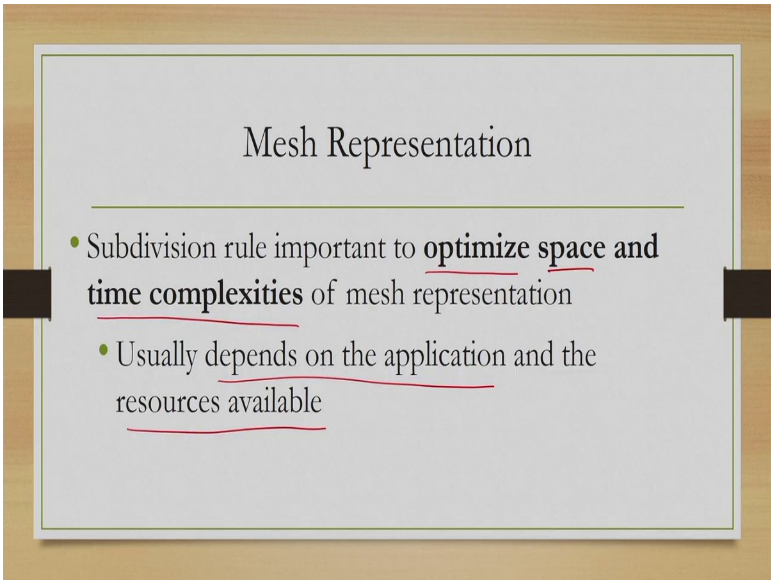
(Refer Slide Time: 8:14)




Because more the number of polygons, the better the approximation is, this is obvious. However, more subdivision also implies more storage and computation. So, if we can use three triangles to represent a surface, which (()) (8:37) if we are using 30 triangles to represent a surface, the latter representation, of course, will give a better visual clarity, better visual quality.

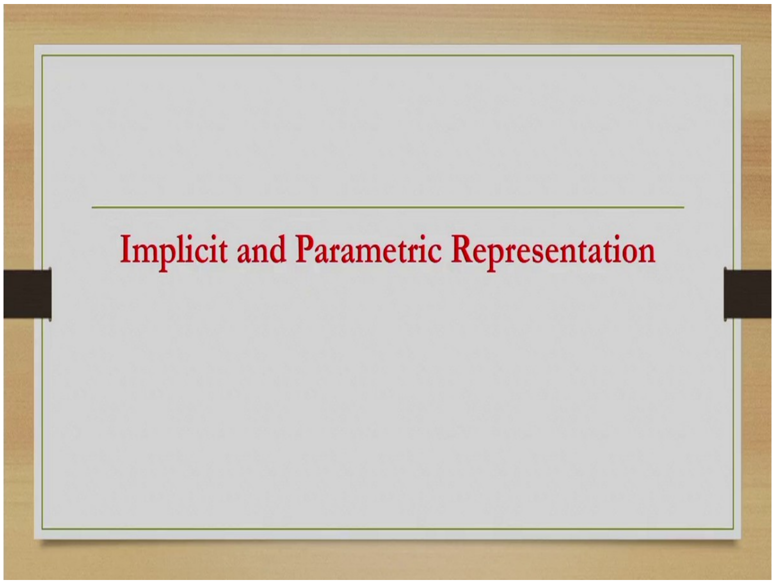
However, since we are increasing the number of objects or polygons in the mesh, there will be a corresponding increase in the storage because we have to now store vertices for 30 triangles, which are (()) (9:08) 3 triangles as well as computations, because we have to perform recursive subdivision to create this mesh, a larger number of times, which (()) (9:19) when we have less number of triangles. So, creation of mesh is computation intensive and storing the mesh information is also storage intensive, and if we increase both, then both needs to be taken into account.

(Refer Slide Time: 9:44)



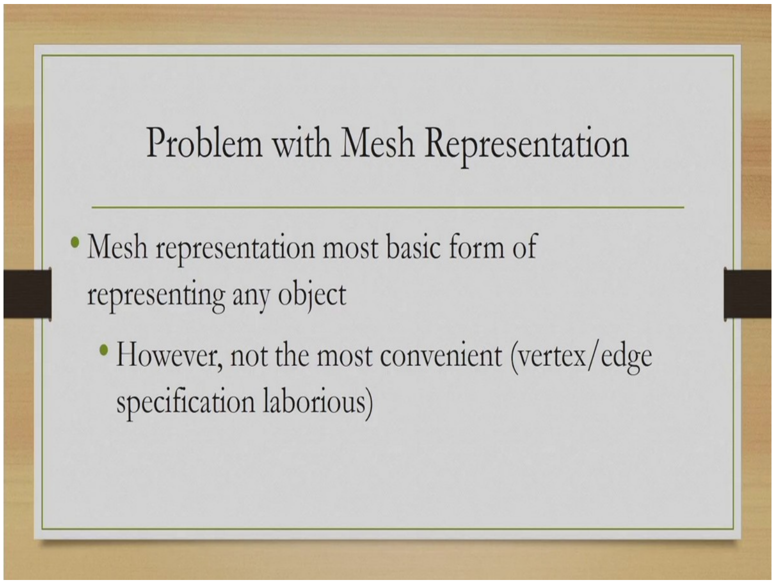

So, there is a trade-off and what we need to do is to optimize space and time complexities while keeping the quality acceptable, quality of representation acceptable. Now how to decide how to balance this tradeoff? The answer depends on the application and the resources available. Depending on the resources and depending on what we need to render we can choose the right value for the number of subdivisions required and as well as the number of polygons. We are going to be to approximate a surface with a mesh. That is about mesh representation.

(Refer Slide Time: 10:37)




Next let us move to the other two representations, implicit and parametric representations.

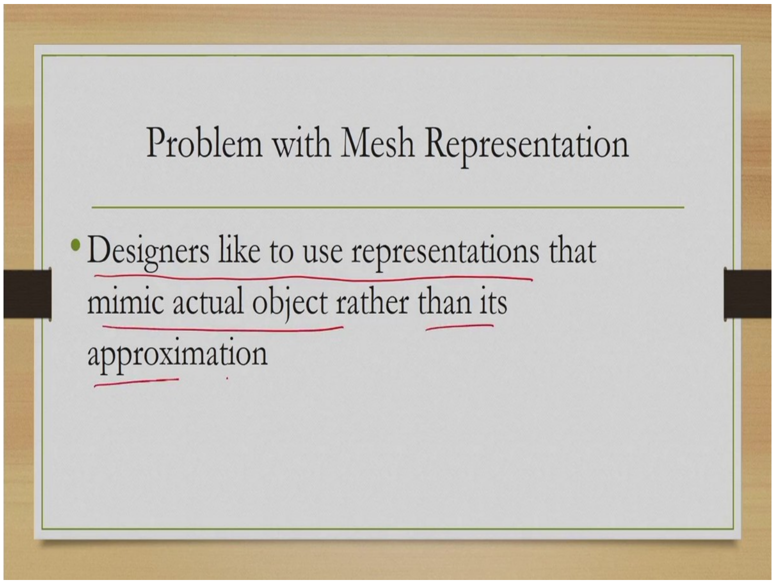
(Refer Slide Time: 10:46)




Now, although we said that mesh representation is the most fundamental type of representation, for a developer it is not necessarily a very convenient mode of representation because for complex surfaces, first of all, it is very difficult to determine how many polygons should be used to create a mesh. Secondly, it is very cumbersome to enumerate all the vertices of the mesh.

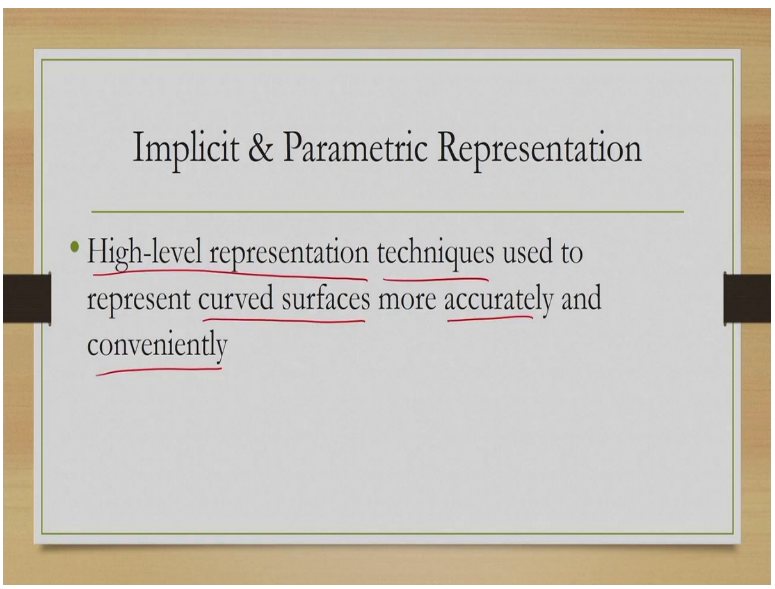
If the number of polygons in the mesh or the number of meshes that we are using are large, which is likely to be the case in any practical application. So, what is required is some compromise and some way to help the developer define objects without bothering too much or spending too much time on defining the meshes.

(Refer Slide Time: 11:52)



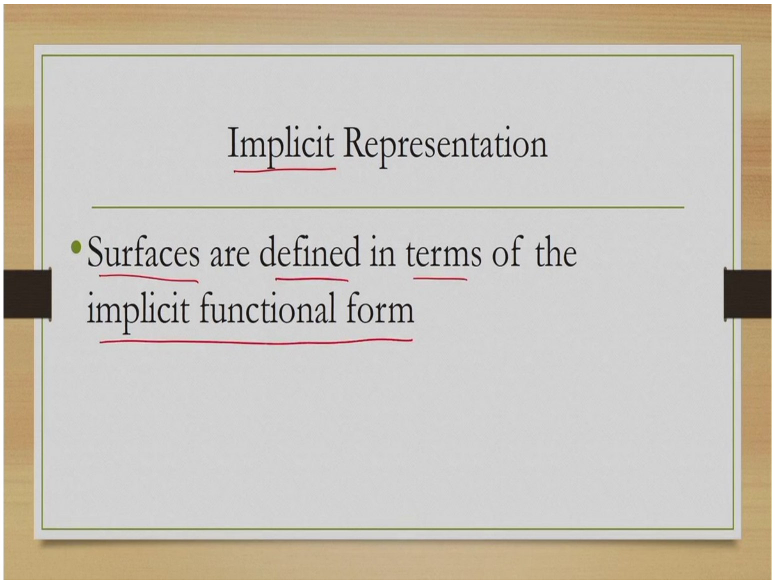

So, designers or developers like to use representations that mimic actual object rather than its approximation.

(Refer Slide Time: 12:04)



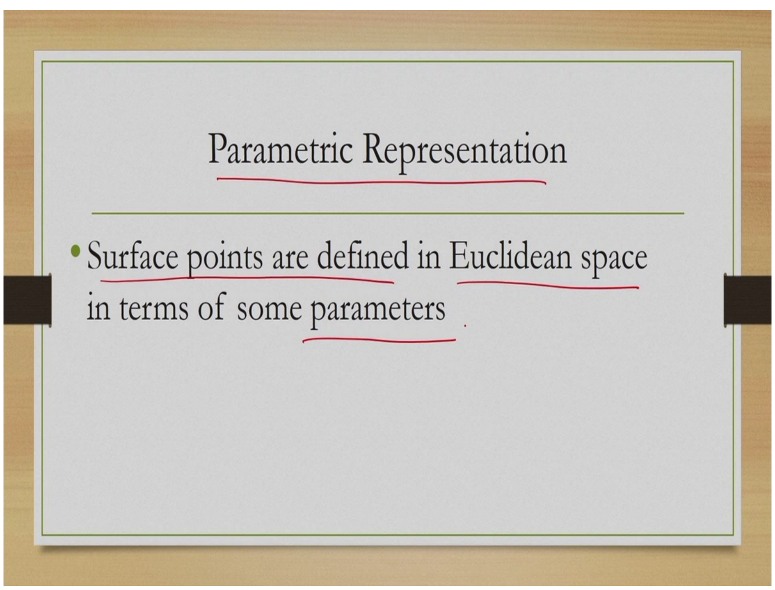

This brings into picture some high level representations, representation techniques, for curved surfaces. Now these techniques are likely to represent curved surfaces more accurately and conveniently for the designer, these are not approximations, rather more closer to the actual representations.

(Refer Slide Time: 12:32)



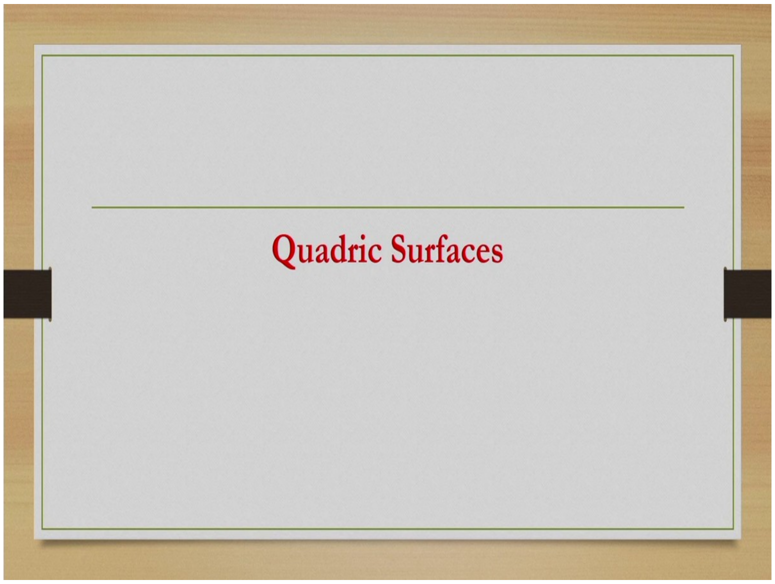

So, implicit and parametric representations are essentially those type of representations where it is more convenient and represents objects in more accurate way rather than approximate the objects. Now, let us start with implicit representation. So, in this case the surfaces are defined in terms of implicit functional form, some mathematical equations.

(Refer Slide Time: 13:05)



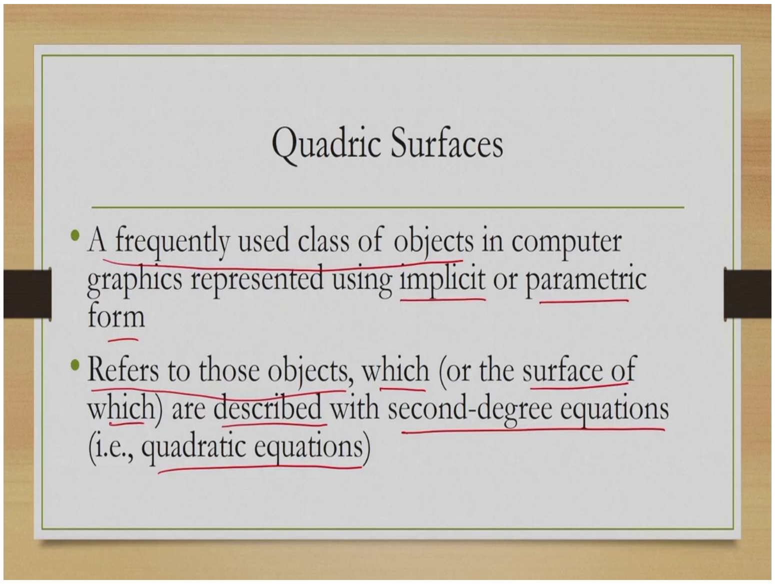

In case of parametric representation, the surface points are defined in Euclidean space in terms of some parameters, again in the form of some mathematical equations.

(Refer Slide Time: 13:23)



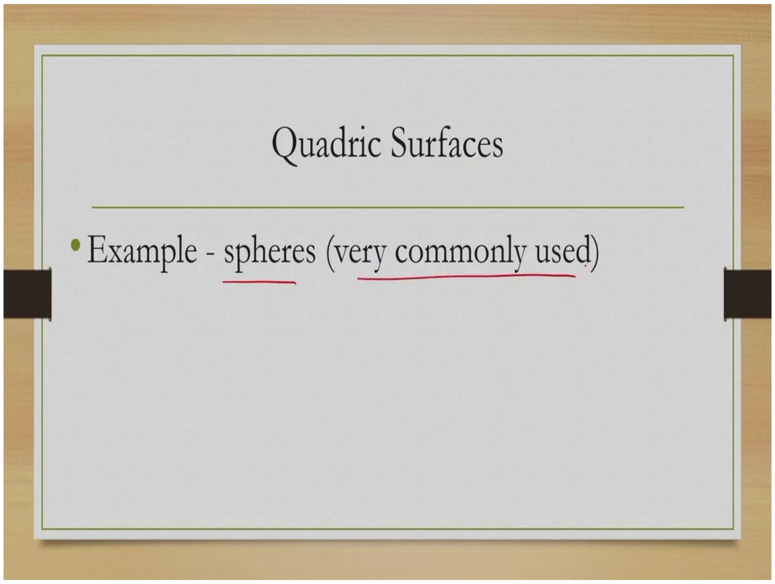

Now, let us see a few examples which are popularly used in graphics. Let us start with quadric surfaces.

(Refer Slide Time: 13:41)



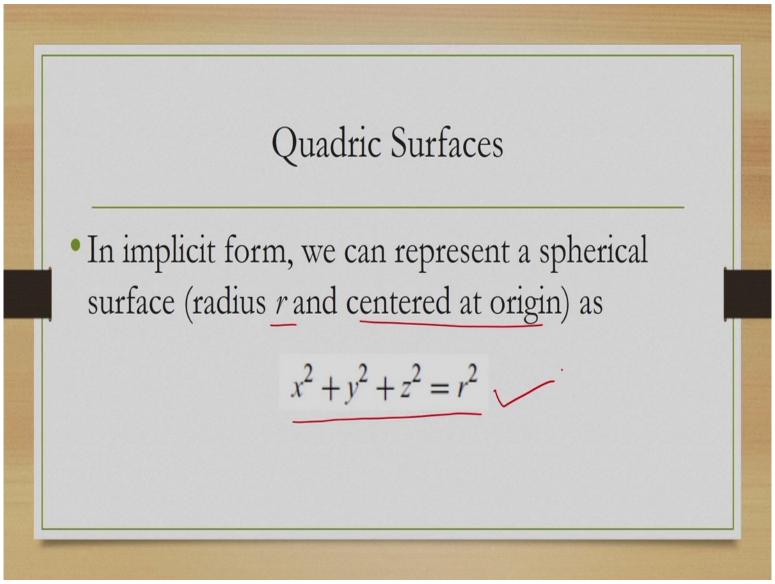

These are frequently used class of objects in graphics which are represented using implicit or parametric form. And this term quadric surfaces refers to those objects, which or the surface of which are described with second degree equations or quadratic equations.

(Refer Slide Time: 14:11)



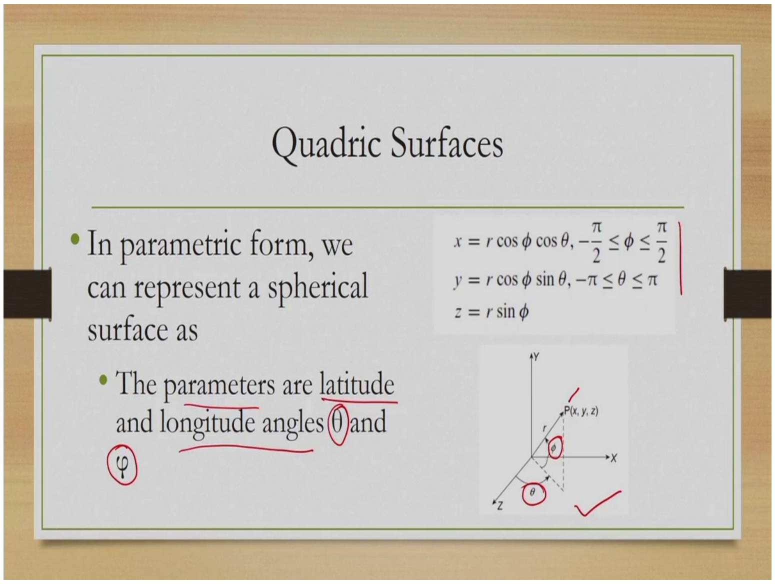

For example, spheres, these are very commonly used.

(Refer Slide Time: 14:19)



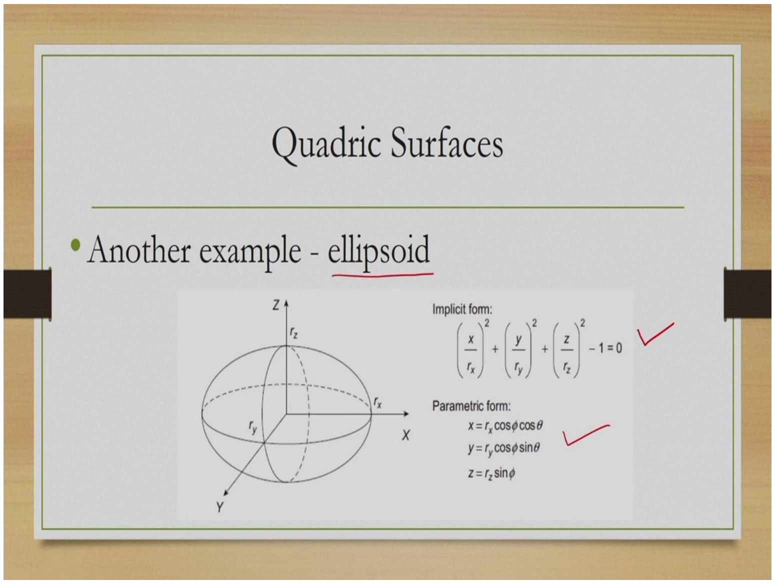

In implicit form, we can represent a spherical surface with radius r and, which is centered at origin as $x^2 + y^2 + z^2 = r^2$. So, this equation we can use for implicitly representing a sphere.

(Refer Slide Time: 14:42)



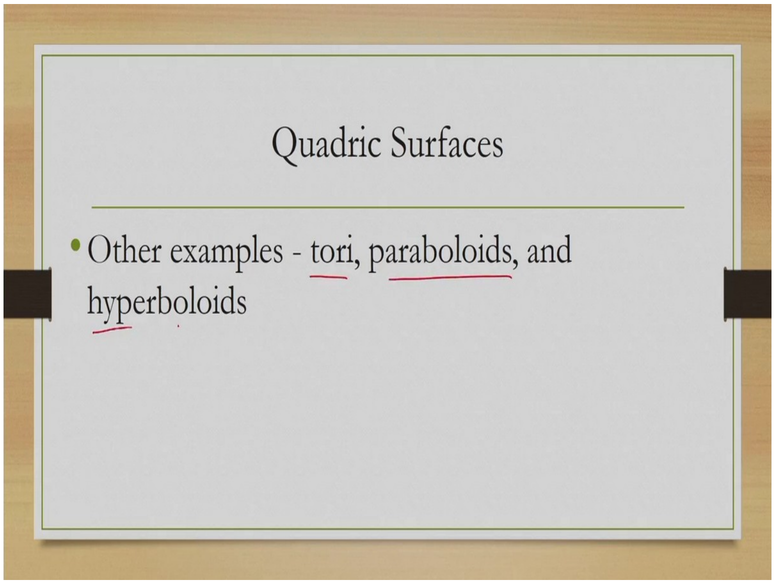

The same sphere can be represented parametrically also using this form where the angles theta and phi of the parameters which represent the latitude and longitude angles as shown in this figure here, this is the latitude angle and this is the longitude angle. And this p is a point on this sphere, which is represented using the parameters.

(Refer Slide Time: 15:25)




Similarly, we can represent ellipsoid also either in implicit form as shown here or in parametric form as shown here. This is another widely used quadric surface.

(Refer Slide Time: 15:44)

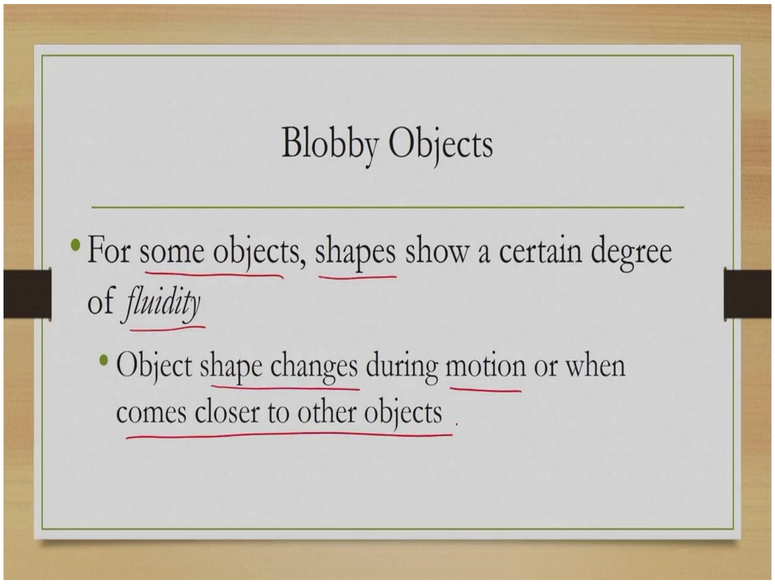



There are many other examples like tori, paraboloids and hyperboloids. Some other widely used quadric surfaces in graphics applications.

(Refer Slide Time: 15:58)

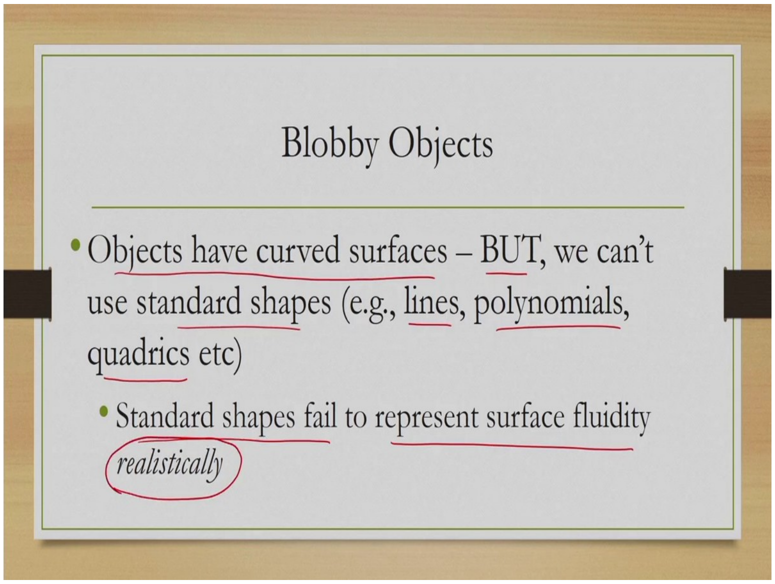

An interesting class of objects are called blobby objects.

(Refer Slide Time: 16:07)



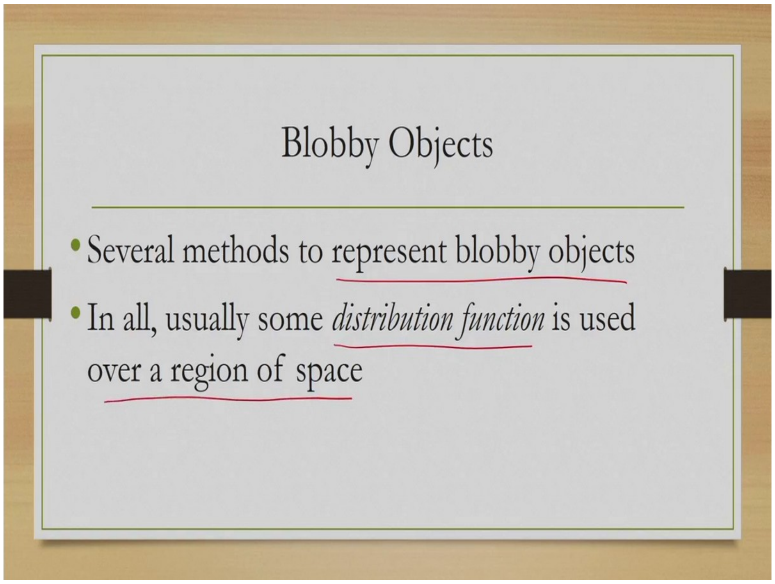

There are some objects for whom their shapes show certain degree of fluidity or flexibility, that means the object shape changes during motion or when comes closer to other objects.

(Refer Slide Time: 16:25)



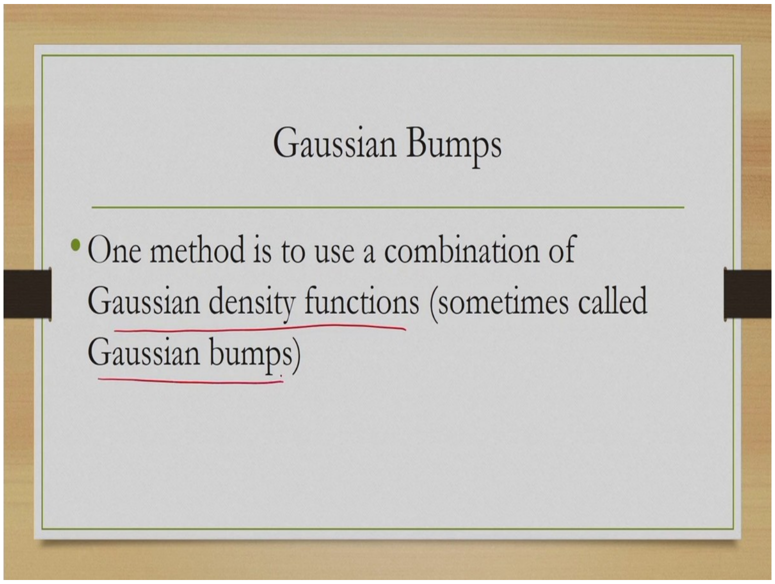

Typically, these objects have curved surfaces, but we cannot use standard shapes like lines, polynomials or quadratics, quadratic equations or quadrics to represent these shapes because these equations or standard shapes fail to represent surface fluidity in a realistic way. So, we have objects which show some fluidity, whose surfaces are represented using some curves, but those curves we cannot represent using line or polynomials or quadrics because then we will lose the fluidic nature.

(Refer Slide Time: 17:11)



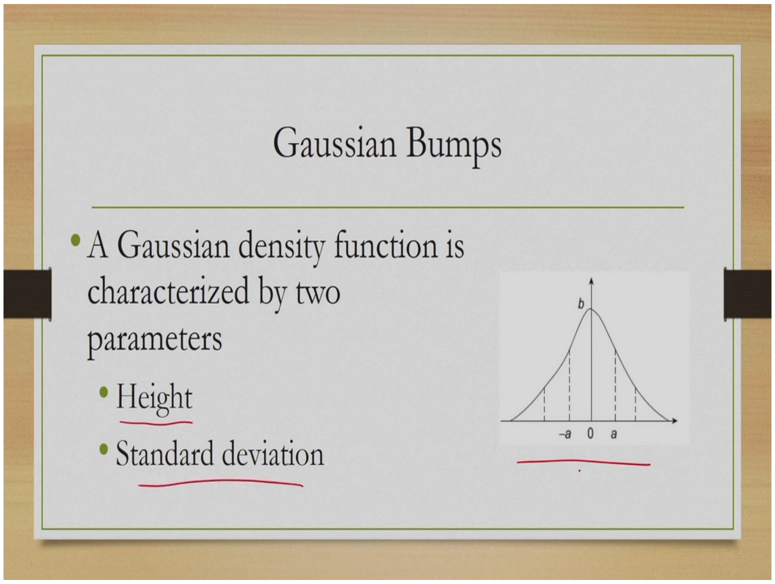

Now such objects generally are referred to as blobby objects such as molecular structures, liquid and water droplets, melting objects, animal and human muscle shapes and so on, these are some examples there are many other examples also. There are several methods to represent blobby objects. In all, there is one common approach essentially to use some distribution function of over a region of space.

(Refer Slide Time: 17:49)



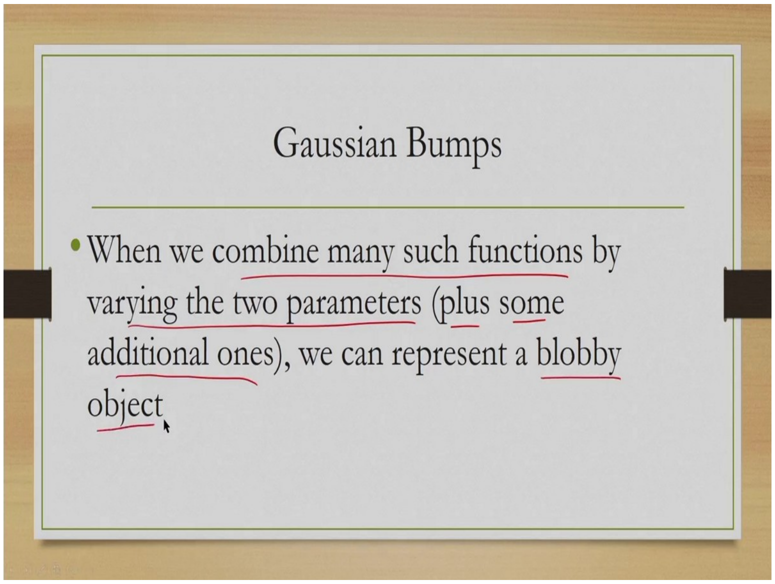
One method is to use a combination of Gaussian density functions or sometimes called Gaussian bumps.

(Refer Slide Time: 17:59)



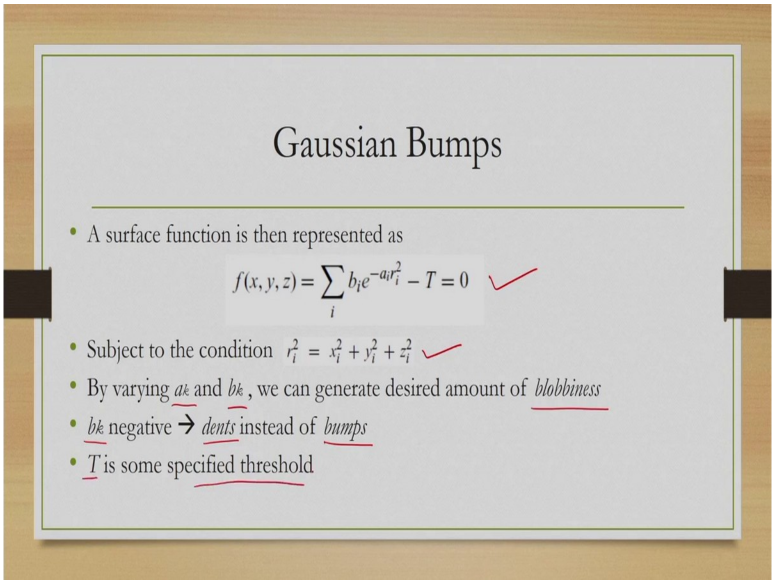

An example is shown here of a Gaussian density function, it is characterized by two parameters, height and standard deviation as shown in the figure.

(Refer Slide Time: 18:19)



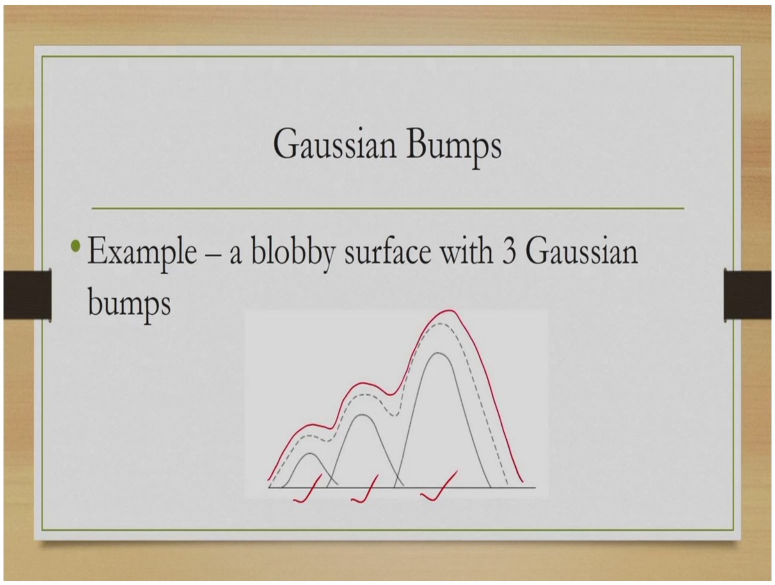

Now, when we combine many such functions by varying the two parameters, plus some other parameters, we get a blobby object or we can represent a blobby object.

(Refer Slide Time: 18:36)



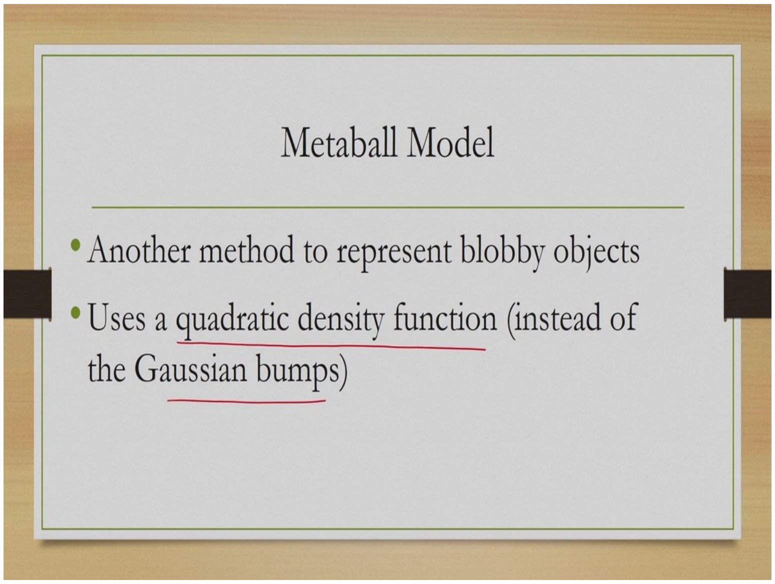

So, the object can be represented with a function like this. Subject to the condition mentioned here. Now by varying the parameters, $a_k$ and $b_k$ we can generate desired amount of blobby-ness or fluidity that we require. Now, when $b_k$ becomes negative, then there are dents instead of bumps and T is some specified threshold.

(Refer Slide Time: 19:12)



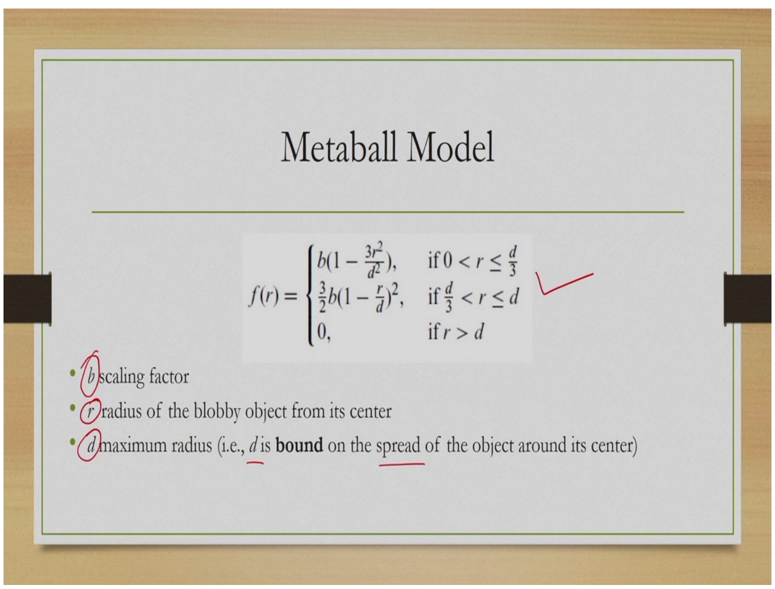

An example is shown here where we have used three Gaussian density functions by varying the parameters to create an overall shape, something like this as shown in this dotted line.

(Refer Slide Time: 19:30)



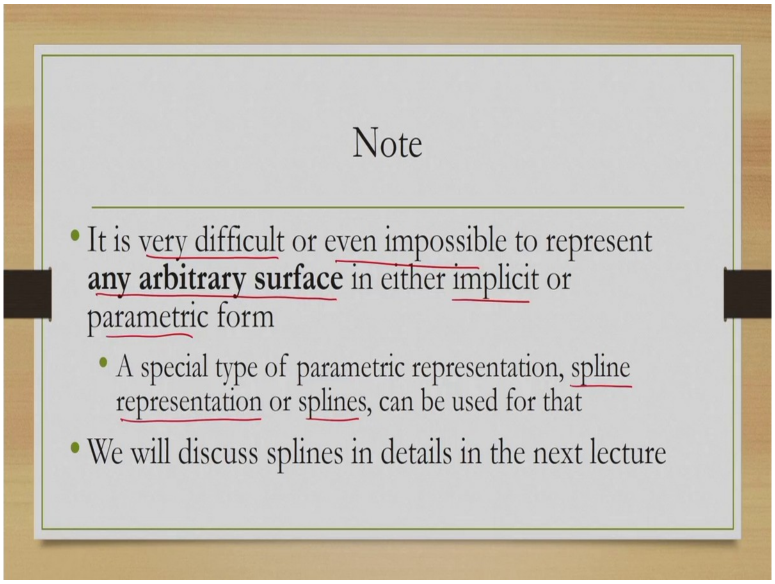

There is another interesting method to use blobby object. This is also quite popular where a quadratic density function instead of Gaussian bumps is used.

(Refer Slide Time: 19:46)




Which looks something like this b is the scaling factor, r is the radius of the object and d is maximum radius, d is the bound on the spread of the object around its center. So how far the object is constrained around the center is specified by d. So, these three are the parameters using which we can define blobby object in this metaball model.

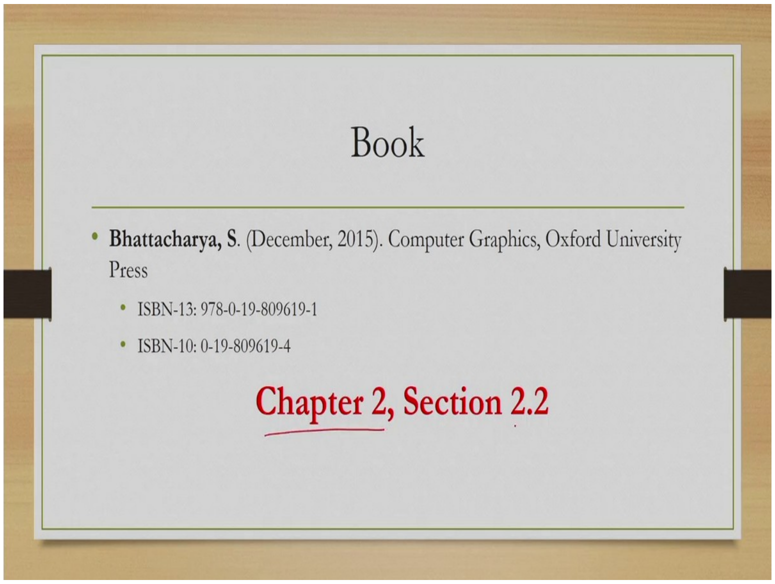
(Refer Slide Time: 20:23)




Now, these are some techniques that we have discussed, however it is very difficult or even impossible to represent any arbitrary surface in either implicit or parametric form. The functions that we have already seen are quite complex in itself. But still there are other surfaces which may turn out to be very difficult, which are indeed very difficult to represent using such equations. So, in order to represent such surfaces, we use a special type of parametric representation called spline representation or splines. Now these splines we will discuss in more details in the next lecture.

So today, we have got an introduction to various boundary representation techniques, so we learned about mesh representation, we learned about basic idea of implicit and parametric representation techniques with some detailed discussion on quadric surfaces and blobby objects. In the next lecture, we will continue our discussion on boundary representation technique next, few lectures will be devoted to a detailed discussion on spline representations that will be followed by a discussion on space partitioning methods. That is all for today.

(Refer Slide Time: 22:02)




So, whatever I have covered today can be found in this book. You are advised to go through Chapter 2, Section 2.2 for the topics that are covered today. We will meet again in the next lecture till then goodbye.